Performance Evaluation of Classifiers used for Identification of Encryption Algorithms

Suhaila Omer Sharif[1], Saad P. Mansoor[2]
[1,2] School of Computer Science, Bangor University, Bangor, UK
[1] s.o.sharif@bangor.ac.uk
[2] s.mansoor@bangor.ac.uk

***Abstract*—Evaluating classifier performance is a critical problem in pattern recognition and machine learning. In this paper pattern recognition techniques were applied to identify encryption algorithms. Four different block cipher algorithms were considered, DES, IDEA, AES, and RC2 operating in (Electronic Codebook) ECB mode. Eight different classification techniques were used for this purpose, these are: Naïve Bayesian (NB), Support Vector Machine (SVM), neural network (MLP), Instance based learning (IBL), Bagging (Ba), AdaBoostM1 (MdaBM1), Rotation Forest (RoFo), and Decision Tree (C4. 5). The result shows that using pattern recognition is a useful technique to identify the encryption algorithm, and according to our simulation using one encryption of key provide better classification than using different keys. Furthermore, increase the number of the input files will improve the accuracy.**

***Index Terms*—Pattern Recognition, Classifiers, Encryption, Cryptanalysis**

I. Introduction

A typical cipher takes a plaintext message and some secret keying data as its input and produces an encrypted version of the original message known as the ciphertext. An attack on a cipher can make use of the ciphertext alone or it can make use of some plaintext and its corresponding ciphertext. Cryptanalysis is the process of recovering the plaintext and/ or key from a ciphertext. Most encryption algorithms have a finite key space and, hence, are vulnerable to an exhaustive key search attack. However, it is very difficult to identify the encryption keys because the size of the key is such that the time and resources required are not generally available. A random search through a finite but large key space is not usually an acceptable cryptanalysts tool. In cryptanalysis when only the ciphertext is obtained, there are initially two significant tasks, identification of the encryption technique applied, and the encryption key identification. Statistical techniques and machine learning based techniques have been applied to identify the encryption method from the encrypted files. In this work, however, we are exploring the possibility of identifying encryption algorithm by applying pattern recognition techniques. The statistical methods use the frequency at which letters of the alphabet occur in the encrypted file. While in machine learning based methods, the task of identification of the encryption method is considered as a pattern classification task. There are a number of research papers that evaluate some of the classification algorithm. Dileep [1] suggests using a Support Vector Machine for Identification of a block cipher. The identification of encryption method was considered as a document categorization task. A.Cufoglu et al. [2] used pattern recognition classifiers to identify user profiling. The results observed that NB classifier generate the best performance over user related information. In [3] a study was carried out to use classifiers to identify the cipher encryption method. The simulation results show that, the RoFo classifier has the highest classification accuracy attributes by using tree structures obligatory on Naïve Bayesian. In this study a simulation test was carried to evaluate different types of classifiers in terms of classification accuracy of the encryption data.

II. Basics of Classifiers

This section briefly explains the classifiers used in this study.

A. Naive Bayesian (NB)

Naïve Bayesian is one of the most computationally straightforward and efficient Bayesian Classifier methods. It has been used as an effective classifier for many years. It has two advantages over many other classifiers. The first being, it is easy to construct, as the structure is given a priori thus, no structure learning procedure is required. Secondly, the classification process is very efficient. This is due to the assumption that all the features are independent of each other. It has surprisingly out performed many sophisticated classifiers, especially where the features are not strongly correlated [4].

B. Support Vector Machines (SVMs)

This is a useful technique for data classification. It is based on statistical learning theory presented by V.N.Vapnik, and has been productively applied to many classification and pattern recognition problems such as text classification and image recognition. These methods do classification by building an N-dimensional hyper plane that optimally divides the data into two groups: the training and test data points. The objective of SVM method is to produce a model (based on the training data) which predicts the target values of the test data knowing only the test data attributes [5].

C. Neural Networks (MLP)

These classifiers are widely used in many fields. They model the concept of neurons in neuroscience and include sets of connected nodes along with the weights for nodes

DOI: 02.ICT.2011.02.60

and arcs. Neural nets are self-organizing, adaptive and nonlinear. Neural network are versatile and powerful classifiers, but they rely on a number of parameter choices to specify the network architecture and to control the training process [5,6].

D. Instance Based Learner (IBL)

IBL generates classification predictions using only specific instances. Unlike Nearest Neighbour algorithm, IBL normalizes its attributes' ranges, processes instances incrementally, and has a simple policy for tolerating missing values. IBL applies a simple Euclidean distance function to supply graded matches between training instances and given test instance [7]. Equation (1) represents the similarity that is used within the IBL algorithm.

$$\text{Similarity}(x, y) = -\sqrt{\sum_{i=1}^{n} f(xi, yi)} \quad (1)$$

Where *n*: is the number of instance attributes

The numeric value attributes are represented by (2).

$$f(xi, yi) = (xi - yi)^2 \quad (2)$$

The Boolean and symbolic $\neq$tributes are represented by (3).

$$f(xi, yi) = (xi \quad yi) \quad (3)$$

E. Bagging (Ba)

This ensemble approach uses a predefined number of classifier, each one trained on a bootstrap sample of training data. Each classifier is trained on a set of *n* training examples, drawn randomly with replacement from the original training set of size *n*. Such a training set is called a bootstrap replicate of the original set. Each bootstrap replicate contains, on average, 63.2% of the original training set, with many examples appearing multiple times. [8, 9].

F. AdaBoost Algorithm Boosting

The most popular classifier is AdaBoost, a powerful classifier built with linear combination of member classifiers. The idea behind it is obtaining a highly accurate classifier by combining many weak classifiers. Each weak classifier is required to be moderately accurate, i.e. better than random guessing. The classifiers in the ensemble are added one at a time so that each subsequent classifier is trained on data, which have been "hard" for the previous ensemble members. A set of weights is maintained across the objects in the data set so that objects that have been difficult to classify acquire more weight, forcing subsequent classifiers to focus on them. The ensemble construction through AdaBoost comes from theory and is, in fact, equivalent to a logistic regression model by a fitting an additive stage-wise estimation procedure. One of the reasons why AdaBoost algorithm yields good results is the diversity among the weak classifiers. Still, there is no specific benchmark to determine diversity in the process [10, 11].

G. RotationForest (RoFo)

This classifier is based on feature extraction and was proposed by Rodriguez et al [10]. To create the training data for a base classifier, the feature set is randomly split into K subsets (K is a parameter of the algorithm) and Principal Component Analysis (PCA) is applied to each subset. All principal components are retained in order to preserve the variability information in the data. Thus, K axis rotations take place to form the new features for a base classifier. The idea behind the rotation approach is to encourage simultaneously individual accuracy and diversity within the ensemble. Diversity promoted through the feature extraction for each base classifier. Decision trees were chosen here because they are sensitive to rotation of the feature axes, hence the name "forest." Accuracy is sought by keeping all principal components and using the whole data set to train each base classifier. A very useful characteristic of the proposed method is that RoFo can be used in conjunction with practically any base classifies that are used in the ensemble creation process. The thought of the rotation approach is to encourage simultaneously individual accuracy and diversity within the ensemble.

H. C4.5 Classification Algorithm

This algorithm constructs a decision tree (DT) starting from a training set, which is a set of cases. Each case specifies values for a collection of attributes and for a class. The decision tree is used to classify (assign) a class value to a case depending on the values of the attributes of the case. The class specified at the leaf is the class predicted by the decision tree. A performance measure of a decision tree over a set of cases is the classification error which is the percentage of the misclassification cases whose predicted class differs from actual class [12].

III. EXPERMINT SETUP

Numbers of experiments were carried out to establish the classifiers accuracy to identifying the following encryption algorithms AES, RC2, DES and IDEA, operating in electronic codebook (ECB) mode. In the simulation, different instances were used (120, 240, and 400); these represent the total number of input files (ciphertext) used. The plaintext was encrypted using 1, 3, 5, and 30 keys for each algorithm. Matlab® was used to extract an 8-bits histogram of the encrypted data; this was then submitted to WEKA (Weikato Environment for Knowledge Analysis) machine learning platform that supplies a work bench consisting of a collection of learning schemes, which are applied for data mining and machine learning. The primary learning schemes in Weka are "classifiers" and they tempt a rule set or decision tree that models the data, in addition Weka has different algorithms for learning association rules clustering data, visualization, regression, and data processing.

IV. SIMULATION RESULTS

In this section the results of eight classifiers are compared, based on the accuracy of the identification. The simulations conducted using four different block cipher algorithms with different key sizes. Each dataset have the same number

DOI: 02.ICT.2011.02.60

of attributes and different number of instances, which varies from 120 to 400 instances. A 10 fold cross-validations was used as a test mode where 10 pair of training sets and testing sets is created. All the classifiers run on the same training sets and have been tested on the same testing sets to obtain the classification accuracy. To estimate a value on training data set, we define an accuracy measure as shown in equation (4).

$$\text{Accuracy (\%)} = \frac{Correctly\_classifed}{Total} * 100 \qquad (4)$$

In the first simulation, we have been using 240 input files with AES (128,192, and 256-bits), DES 64-bit, IDEA 128-bit, and RC2 (42, 84, and128-bits) where each algorithm has 30 input files. The simulation was conducted using different numbers of keys for each algorithm; Fig 1 shows the calculated accuracy for different set of data. It can be seen that, using 1 encryption key produces high classification accuracy (100%) meaning that the 240 instances (files) were correctly classified. In contrast using 30 different encryptions keys (one for each file) result in lower classification accuracy. This is expected as the keys are generated randomly and will affect the pattern of test data. The results show that the accuracy of the classification will be reduced with increased the number of the encryption keys. The second simulation, deals with the effect of increasing the number of input files (instances) on overall accuracy. Here for each algorithm, AES 128-bits, DES 64-bits, IDEA 128-bits, RC2 128-bits we used 30, 60 and 100 input files, thus the total numbers of the instances are 120, 240 and 400 respectively. Fig 2 shows that for all classifiers using 400 input files achieve high accuracy and using 100 input files achieved lowest accuracy. It is also evident that RoFo classifier produces the best results for all instances, and IBL perform very badly when operating with 100 and 240 instances. In final simulation the effect of the number of keys used to encrypt the plaintext was investigated for the following encryption algorithm AES, RC2, and IDEA with fixed 128-bit encryption key and DES with 64-bits. The number of input files used was 30 and the number of keys was 1, 3, 5 and 30. Here, Fig 3 shows that RoFo classifier has better overall performance and AdaBM1 achieved lowest accuracy. Furthermore, and as expected all the classifiers (apart from AdaBM1) produce good accuracy when using 1, 3 and 5 encryption keys. It can be seen as expected that the accuracy drops to 25 –55 when using one key per each file (30 keys).

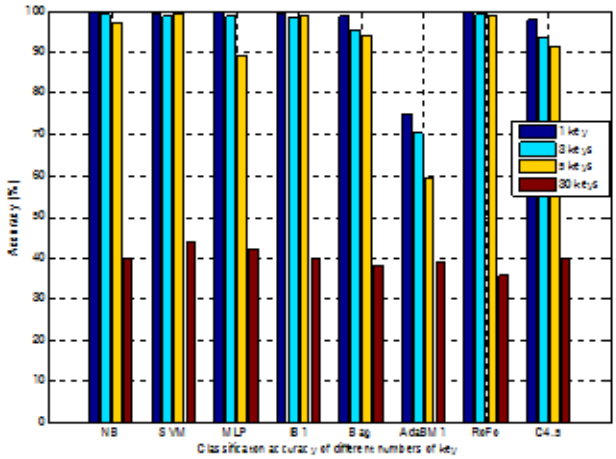


Figure 1. Classification accuracy for each algorithm

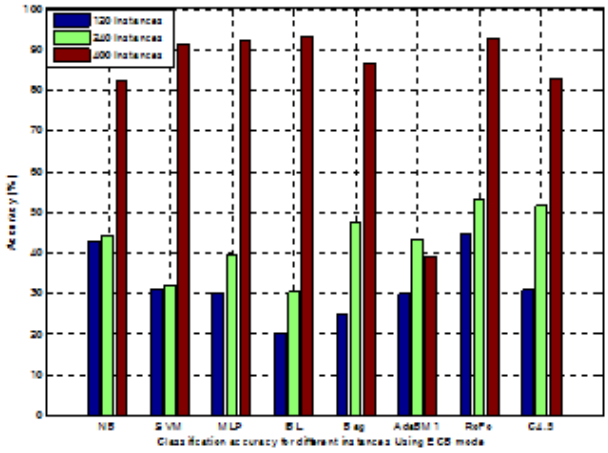


Figure 2. Classification accuracy for each algorithm

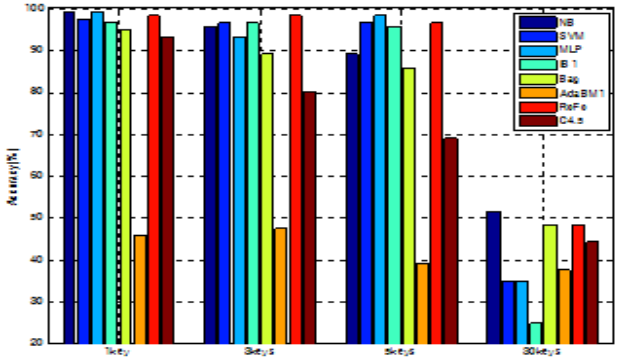
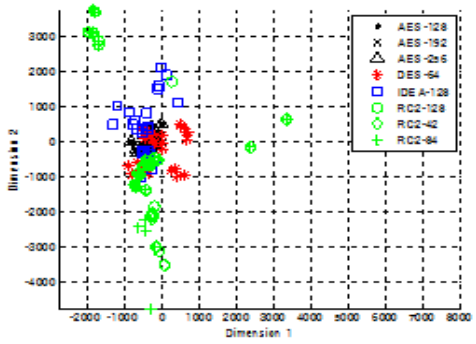
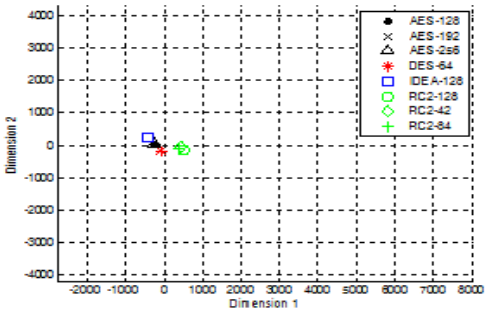


Figure 3. Encryption accuracy for different key seizes



(a) All data



(b) Class centers

Figure 4. Scatter plot for 240 data points

DOI: 02.ICT.2011.02. 60

Fig 4, shows data scatter-plots and the centres of the 240 data points. The plot shows that the classes are founded sporadically. The centres of the "clouds" of points for 8 classes were plotted in Fig 4-b, according to this plot, all algorithms have similar representation. Note that the scales of all plots are different. The class centres are indistinguishable if plotted on the axes of a subplot. Finally, we can say this highlights the difficulty in recognising the type of code through simple pattern classification methods. Fig 5 shows an image of the distance matrix D computed by using identification accuracy of each encryption algorithm. The block of 30 by 30 distances is outlining with black lines. Blue color means high similarity while yellow and red indicate low similarity. The class labels are as follows: 1 AES(128), 2 AES(192), 3 AES(256), 4 DES(64), 5 IDEA(128), 6 RC2(128),7 RC2(42), and 8 RC2(84). The encoding technique that stands out from the rest is AES. The 3-by-3 block sub-matrix in the top left corner is largely blue, shown the similarity within the code. Interestingly, AES algorithm is distinguishable from the rest of the algorithm also noted that the three versions of AES (128, 192, and 256) are not distinguishable within AES. The red vertical and horizontal lines demonstrate the unusually large distances compared to the rest. Unlike ECB, there is no clear pattern to suggest that any of the codes are distinguishable.

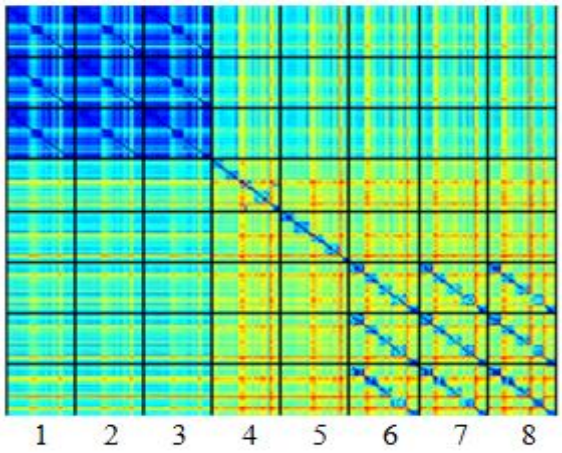


Figure 5. Distance matrix D

Conclusions

In this paper, a simulation experiment was conducted to identify encryption algorithm of encrypted data using variety of classifies. The results show that pattern recognition techniques, are useful tools for cryptanalysis as a means of identifying the type of encryption algorithm used to encrypt the data. This work shows that increasing the number of encryption keys will result in reducing the classification accuracy. Nevertheless, the result shows that it is possible to achieve accuracy above 40% with some classifiers when each file was encrypted with different key. It was also clear that increase the number of files used will improve accuracy.

References

[1] A. Dileep and C. Chandra Sekhar, "Indemnification of Block Ciphers using Support Vector Machines, "International Joint Conference on Neural Networks" Vancouver, Canada, pp. 2696–2701, July 16-21,2006.

[2] A. Cufoglu, M. Lohi, K. Madani, "A Comparative Study of Selected Classifiers with Classification Accuracy in User Profiling", World Congress on Computer Science and Information Engineering. Pp 708-712, San Francisco, 2009.

[3] S.O. Sharif, L.I. Kuncheva, and S.P. Mansoor, "Classifying Encryption Algorithms Using Pattern Recognition Techniques," Neural Networks, 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 2010

[4] A.Cufoglu, M. Lohi, and K. Madani, "Classification Accuracy Performance of Naïve Bayesian (NB), Bayesian Networks (BN), Lazy Learning of Bayesian Rules (LBR) and Instance-Based Learner (IB1) - comparative study," 2008 International Conference on Computer Engineering & Systems, 2008, pp. 210-215.

[5] L. I. Kuncheva., J. J. Rodríguez, "Classifier ensembles for fMRI data analysis: an experiment," Magnetic resonance imaging, vol. 28, no. 4, pp. 583-93, May. 2010.

[6] Y. Maeda and M. Wakamura, "Simultaneous Perturbation Learning rule for Recurrent neural Networks and its FPGA implementation.," IEEE transactions on neural networks a publication of the IEEE Neural Networks Council, vol. 16, 2005, pp. 1664-1672.

[7]David W. Aha, Kibler D. and K. Albert M. (1991) " Instance-based learning algorithms" . Machine Learning journal, Vol 1, No 6, ISDN 1573- 0565, pp. 37-66. [Online] Available from: http://www.springerlink.com/content/kn127378pg361187/fulltext.pdf

[8] Q. He, F. Zhuang, X. Zhao, and Z. Shi, "Enhanced Algorithm Performance for Classification Based on Hyper Surface using Bagging and Adaboost," 2007 International Conference on Machine Learning and Cybernetics, 2007, pp. 3624-3629.

[9] S. Ruggieri, "Efficient C4.5" IEEE Transactions on Knowledge and Data Engineering, vol. 14, 2002, pp. 438-444.

[10] J.J. Rodríguez, L.I. Kuncheva, and C.J. Alonso, "Rotation forest: A new Classifier Ensemble method," IEEE transactions on pattern analysis and machine intelligence, vol. 28, 2006, pp. 1619-1630.

[11] T.-K. An and M.-H. Kim, "A New Diverse AdaBoost Classifier," 2010 International Conference on Artificial Intelligence and Computational Intelligence, pp. 359-363, Oct. 2010.

[12] G. Stiglic and P. Kokol, "Effectiveness of Rotation Forest in Meta- learning Based Gene Expression Classification," Symposium a Quarterly Journal in Modern Foreign Literatures, 2007.

DOI: 02.ICT.2011.02.60